Алексей Васильев. Числовые расчеты в Excel

Книга посвящена методам решения вычислительных задач с помощью Excel: алгебраические уравнения и системы, интерполирование и аппроксимация функциональных зависимостей, дифференцирование и интегрирование, решение дифференциальных и интегральных уравнений. В книге также описываются основные приемы работы с Excel, обсуждаются способы организации документов, анализируются методы ввода и редактирования данных, изучаются возможности применения форматов и стилей, иллюстрируются принципы использования встроенных вычислительных утилит, а также даются основы программирования на VBA. Ранее по теме я опубликовал Вильям Дж. Орвис. Excel для ученых, инженеров и студентов.

Алексей Васильев. Числовые расчеты в Excel. — СПб.: Издательство «Лань», 2022. — 600 с.

Купить книгу в Ozon или Лабиринте

Главная идея книги — это адаптация стандартных (или «классических») алгоритмов решения задач вычислительной математики для реализации в приложении, имеющем «ячеечную» структуру, т.е. структуру электронной таблицы. Автор использует среду Excel 2013, самой свежей версии на момент выпуска книги. Я же использую Excel 365.

Глава 5. Методы вычислений и обработка данных

Числовые формулы и циклические ссылки

Эффектный механизм вычислений основан на использовании циклических ссылок. Циклическая ссылка – это ссылка в формуле на ячейку, которая содержит формулу (т.е. ссылка в ячейке на эту же ячейку). По умолчанию такая ситуация считается ошибочной, поскольку означает, что для вычисления значения ячейки по формуле с циклической ссылкой необходимо использовать значение этой ячейки. Круг замкнулся!

Для использования циклических ссылок пройдите *Файл –> Параметры –> Формулы*. Устанавливаем флажок *Включить итеративные вычисления*.

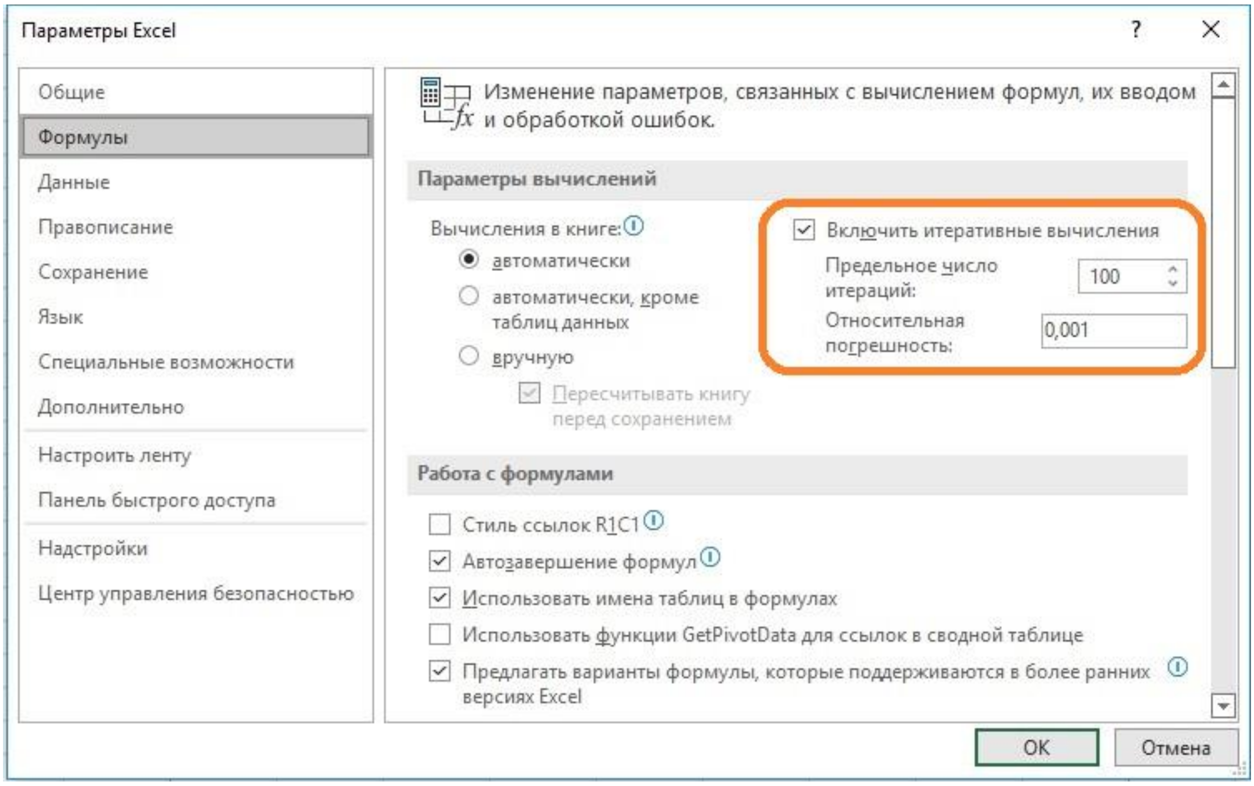

Рис. 1. Переход в режим использования циклических ссылок (итеративных вычислений)

В итерационном процессе начальное значение ячейки всегда равно нулю. Это нулевое значение используется в циклической ссылке для вычисления значения ячейки для первой итерации. Значение ячейки для первой итерации используется для вычисления значения ячейки для второй итерации и т.д. Предельное количество итераций определяется значением поля *Предельное число итераций*. Вычисления также заканчиваются, если относительная погрешность вычислений становится меньше того значения, что указано в поле *Относительная погрешность*.

Метод последовательных итераций может применяться для решения алгебраических уравнений вида x = f(x). Например, найдем корни уравнения $x^2 - 6x + 5 = 0$. У него два корня x = 1 и x = 5.

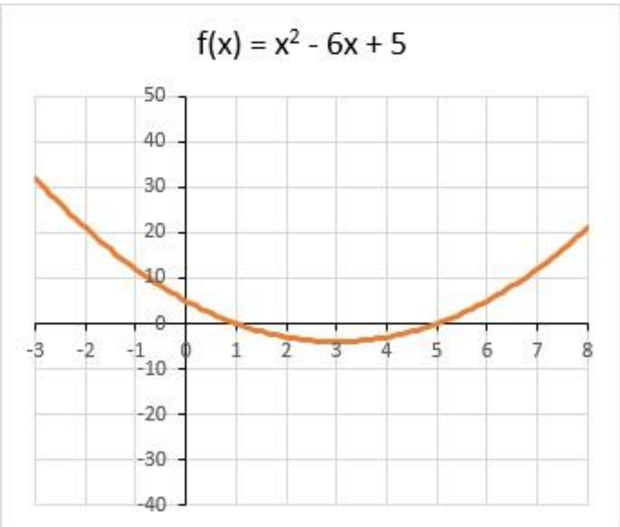

Рис. 2. График функции $y = x^2 - 6x + 5$

Представим уравнение в виде

$$(1)\ x = \frac{x^2 + 5}{6}$$

В ячейку А1 введите формулу =(A1^2+5)/6

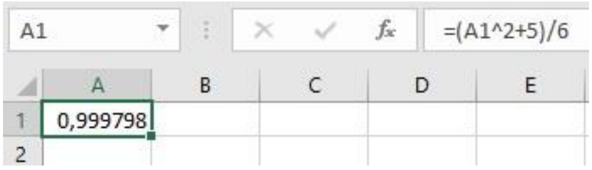

Рис. 3. Нахождение корня уравнения итерационным методом

Точность решения определяется настройками Параметров Excel (см. рис. 1). Если точности не хватило, нажмите <F9>. Итерационный процесс продолжится с текущего числового значения ячейки.

Одна из проблем итерационного процесса заключается в том, что он всегда стартует с нулевого значения в ячейке. Например, чтобы найти второй корень уравнения, его следует представим в виде…

(2) $x = \sqrt{6x - 5}$

Но… при x = 0 под корнем отрицательное значение. Поэтому если в ячейку A1 ввести формулу =КОРЕНЬ(6*A1-5), получим сообщение об ошибке:

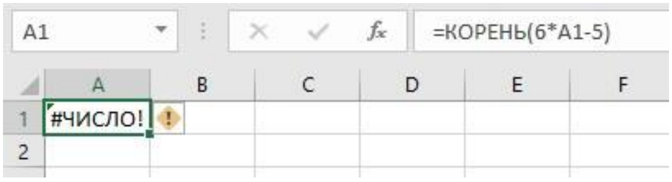

Рис. 4. Ошибка из-за того, что начальное выражение под корнем отрицательное

Изменим начальное значение. Идея состоит в том, чтобы ввести новую переменную z = x – a или x = z + a. И решить новое уравнение z = f(z + a) – a. Для переменной z начальное значение по-прежнему нулевое. В ячейку A2 введите начальное приближение для корня, а в ячейку A1 – формулу =КОРЕНЬ(6*(A1 +A2)-5)-A2.

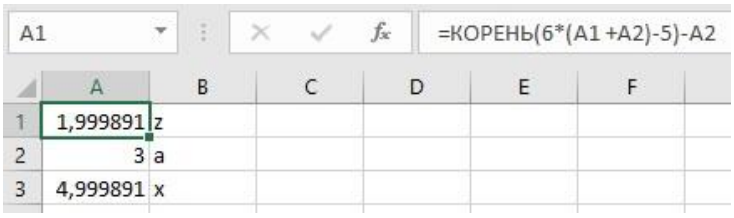

Рис. 5. Начальное значение для итерационных вычислений задается в явном виде

В результате решения уравнения z = f(z + a) - a находим корень для z. При этом x = z + a.

Чтобы метод итераций работал нужно, чтобы в области поиска корня функции f(x) ее производная по модулю была меньше единицы.

(3) $|f'(x)| < 1$

Как минимум, это условие должно выполняться в точке начального значения поиска корня.

Утилита подбора параметра

Большинство расчетных возможностей Excel реализуется через встроенные функции. Но иногда удобнее использовать специальные утилиты или надстройки. Так, например, назначение утилиты *Подбора параметра* состоит в том, чтобы подобрать значение в какой-то определенной ячейке такое, что значение в другой ячейке примет нужное значение.

Чтобы запустить утилиту пройдите *Данные –> Прогноз –> Анализ "что если" –> Подбор параметра*. Решим с помощью этой утилиты рассмотренное выше уравнение.

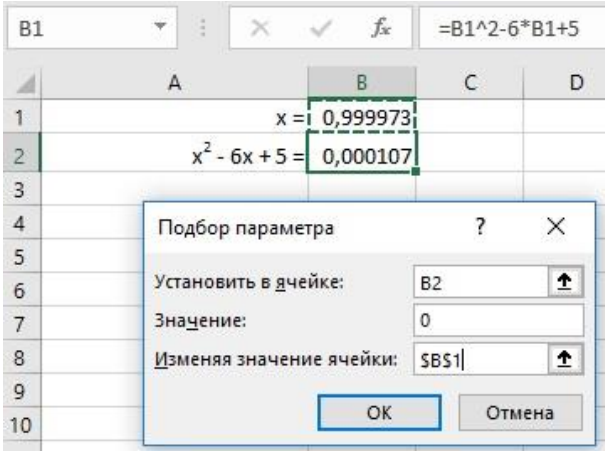

Рис. 6. Решение уравнение с помощью утилиты *Подбор параметра*

Найден один из двух корней уравнения. Второй корень x = 5 можно найти, если изменить начальное значение в ячейке B3. Например, на 8. В реальной жизни угадать, как начальное приближение влияет на результат подбора параметра, бывает крайне сложно. Еще одна проблема заключается в том, что варьируется всего одна ячейка. В этом отношении более гибкой является надстройка *Поиск решения*, используемая в оптимизационных задачах.

Глава 7. Избранные вычислительные задачи

Решение дифференциального уравнения

Дифференциальным называется уравнение вида $F(x, y, y', y'', \dots y^{(n)}) = 0$, связывающее аргумент x, неизвестную функцию $y(x)$ и ее производные. Порядок дифференциального уравнения определяется порядком старшей производной. Задача – найти функцию.

Рассмотрим уравнение первого порядка $y'(x) = f(x, y(x))$. Чтобы найти решение уравнения в числовом виде, необходимо указать начальное условие – значение функции $y(x)$ в точке x_0, т. е. указать числовое значение y_0 такое, что $y(x_0) = y_0$. Дифференциальное уравнение и начальные условия – это задача Коши. Для решения дифференциального уравнения используем не самый точный, но весьма простой метод Эйлера. Чтобы найти значение функции $y(x)$ интервал значений аргумента от x_0 до x разбивается на N одинаковых частей так, что $x_k = x_0 + k\Delta x$, где $\Delta x = (x - x_0)/N$. $k = 0, 1, \dots, N$.

Для вычисления значения $y(x_N)$ последовательно вычисляем значения $y(x_k)$ в узловых точках x_k. При этом используем рекуррентное соотношение $y_{k+1} = y_k + \Delta x * f(x_k, y_k)$ с начальным приближением $y_0 = y(x_0)$. Эту итерационную процедуру реализуем в Excel для уравнения $y'(x) = y(x)*\sin x$ с начальным условием $y(\pi/2) = 10$. Эта задача имеет «точное» аналитическое решение

$$(4)\ y(x) = 10 \cdot e^{-cosx}$$

	A	B	C	D	E	F
1	Номер узла	x	f(x)	Точное решение	Разность	Шаг
2	0	1,570796	10	10	0	0,001
3	1	1,571796	10,01	10,010005	0,000005	
4	2	1,572796	10,02001	10,02002	0,000010	
5	3	1,573796	10,03003	10,030045	0,000015	
6	4	1,574796	10,04006	10,04008	0,000020	
57	55	1,625796	10,56483	10,56511323	0,000282	
58	56	1,626796	10,57538	10,57566734	0,000287	
59	57	1,627796	10,58594	10,5862314	0,000293	
60	58	1,628796	10,59651	10,59680542	0,000298	
61	59	1,629796	10,60709	10,60738938	0,000303	
62	60	1,630796	10,61767	10,61798328	0,000308	

Рис. 7. Результат решения дифференциального уравнения в числовом виде

Здесь в столбце А — номер узла; В — значение аргумента в узле; С — значение искомой функции; D — точное решение уравнения; Е — разность точного и приближенного решений. В ячейке F2 указано значение для шага приращения по аргументу. Формулы см. в приложенном Excel-файле. Часть строк на рис. 7 скрыта.

Вычисление интегралов

Рассмотрим задачу о вычислении интеграла вида

$$(5)\ y(x) = \int_{x_0}^{x} f(z)dz$$

Для интегрирования используем метод трапеций. Интервал интегрирования разбиваем на N частей и выбираем узловые точки $x_k = x_0 + k\Delta x$, где $\Delta x = (x - x_0)/N$, а индекс $k = 0, 1, \ldots, N$. Воспользуемся тем обстоятельством, что числовое значение интеграла равно площади под графиком функции на интервале интегрирования. Разобьем область под кривой на трапеции, с основаниями в узловых точках. Ребро трапеции в точке x_k равняется $f(x_k)$. Тогда площадь области под кривой равняется (примерно) сумме площадей таких трапеции. А площадь одной трапеции равна

$$(6)\ S = \frac{\Delta x(f(x_k) + f(x_{k+1}))}{2}$$

Таким образом, в качестве оценки для значения интеграла получаем выражение

$$(7)\ S = \frac{\Delta x}{2}\sum_{k=0}^{N-1}(f(x_k) + f(x_{k+1}))$$

Воспользуемся этой формулой для вычислений в Excel интеграла

$$(8)\ \int_{0}^{x} \sin(z)\, dz = 1 - \cos(x)$$

	A	B	C	D	E	F
1	Номер уз.	x	Интеграл	Точное решение	Разность	Шаг dx
2	0	0	0	0	0	0,15708
3	1	0,15708	0,012286334	0,012311659	2,53E-05	
4	2	0,314159	0,048842806	0,048943484	0,000101	
5	3	0,471239	0,108769275	0,108993476	0,000224	
6	4	0,628319	0,190590151	0,190983006	0,000393	
7	5	0,785398	0,292290733	0,292893219	0,000602	
8	6	0,942478	0,411366816	0,412214748	0,000848	
9	7	1,099557	0,544886351	0,5460095	0,001123	
10	8	1,256637	0,689561644	0,690983006	0,001421	
11	9	1,413717	0,841830309	0,843565535	0,001735	
12	10	1,570796	0,997942986	1	0,002057	
13	11	1,727876	1,154055664	1,156434465	0,002379	
14	12	1,884956	1,306324329	1,309016994	0,002693	
15	13	2,042035	1,450999621	1,4539905	0,002991	
16	14	2,199115	1,584519156	1,587785252	0,003266	
17	15	2,356194	1,703595239	1,707106781	0,003512	
18	16	2,513274	1,805295822	1,809016994	0,003721	
19	17	2,670354	1,887116698	1,891006524	0,00389	
20	18	2,827433	1,947043166	1,951056516	0,004013	
21	19	2,984513	1,983599639	1,987688341	0,004089	
22	20	3,141593	1,995885973	2	0,004114	

Рис. 8. Результат вычисления интеграла (формулы см. в приложенном Excel-файле)

Глава 8. Диаграммы

Параметрическая кривая

Под параметрической подразумевается кривая, уравнение которой задано в параметрическом виде. Идея определения в параметрическом виде зависимости *y* от *x* состоит в том, что задается зависимость каждой из этих переменных от параметра t. А если заданы зависимости $x = f_1(t)$ и $y = f_2(t)$, то система из этих двух уравнений определяет (неявно) зависимость $y(x)$. Такая зависимость называется параметрической. Если в системе уравнений $x = f_1(t)$ и $y = f_2(t)$ путем алгебраических преобразований удается исключить параметр t, получают явное соотношение между переменными x и y. Для построения графика для зависимости $y(x)$ нужно предварительно вычислить координаты точек (x_k, y_k) ($k = 0, 1, 2, ..., n$). Это делают, используя соотношения: $x_k = x(t_k) = f_1(t_k)$ и $y_k = y(t_k) = f_2(t_k)$.

Мы построим график функции, которая в полярных координатах задана соотношением $r = \sin(5\varphi)$. Здесь r и φ – полярные координаты – радиус и полярный угол. Полярный угол φ может принимать значения от 0 до 2π, а радиус должен быть неотрицательным. Полярные координаты связаны с декартовыми x и y соотношениями $x = r\cos(\varphi)$ и $y = r\sin(\varphi)$. В роли параметра выступает φ. Действительно, декартовы координаты точек на кривой определяются соотношениями $x(\varphi) = r(\varphi)\cos(\varphi)$ и $y(\varphi) = r(\varphi)\sin(\varphi\ \phi)$, где $r(\varphi) = \sin(5\ \varphi)$. Если результат, возвращаемый формулой $r(\varphi) = \sin(5\ \varphi)$, меньше нуля, то присваиваем r значение 0 (r по определению неотрицательна).

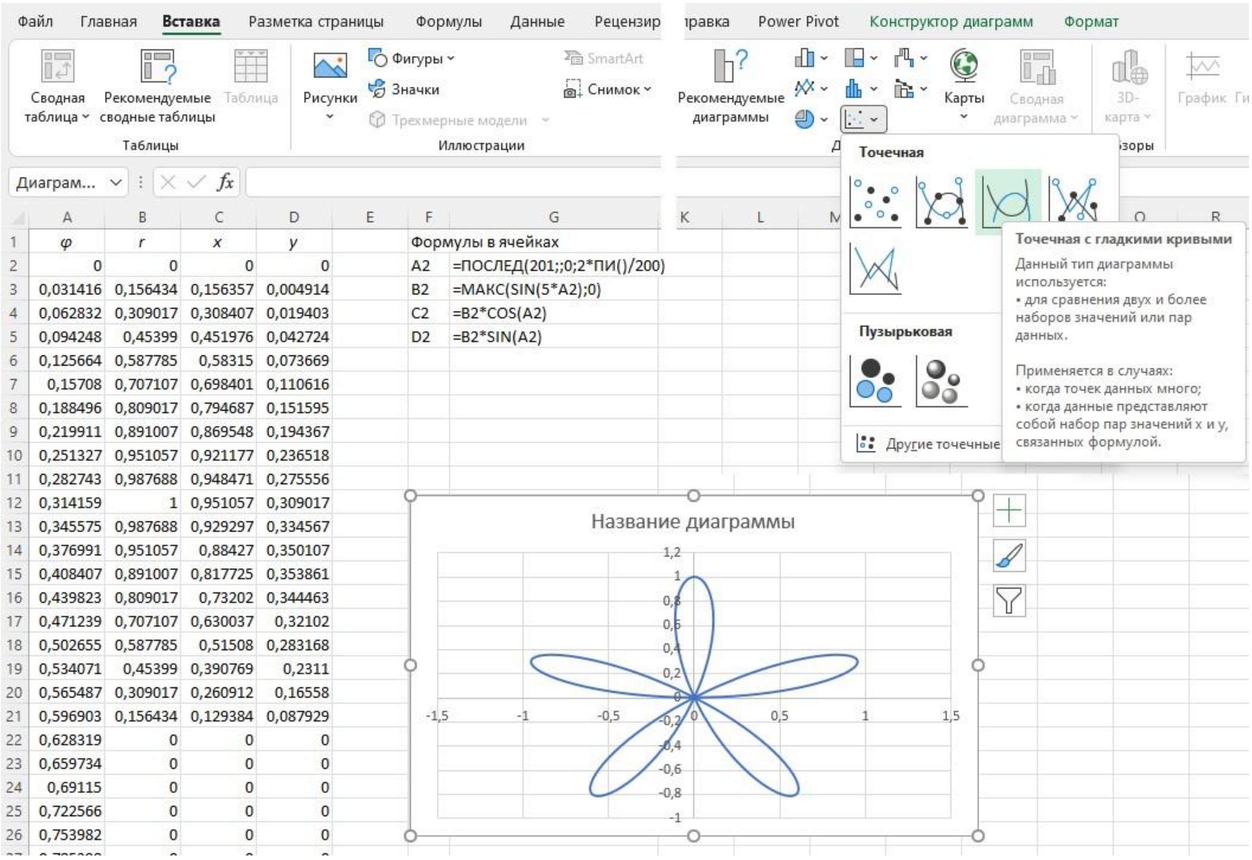

Рис. 9. Параметрическая кривая

Выделите диапазон C2:D202, пройдите *Вставить –> Диаграммы –> Точечная* и выберите тип *Точечная с гладкими кривыми*.

Глава 10. Основы программирования в VBA

Код процедуры, демонстрирующий использование конструкции With … End With

```
Sub GetInfo()
' Размер шрифта ячейки
Dim FSize As Integer
' Название шрифта
Dim FName As String
' Ширина и высота ячейки
Dim CWidth As Integer, CHeigh As Integer
' Соотношение сторон ячейки
Dim CScale As Double
' Текущая дата
```

```
Dim Today As Date
' Текст для отображения в окне
Dim txt As String
' Обработка активной ячейки
With ActiveCell
' Считывание названия шрифта
FName = .Font.Name
' Считывание размера шрифта
FSize = .Font.Size
' Считывание ширины ячейки
CWidth = .Width
' Считывание высоты ячейки
CHeight = .Height
' Вычисление соотношения сторон ячейки
CScale = CWidth / CHeight
' Определение текущей даты
Today = Date
' Формирование текста для отображения в окне
txt = "Сегодня " & Today & vbLf & vbLf
txt = txt & "Параметры активной ячейки:" & vbLf & vbLf
txt = txt & "Шрифт - " & FName & vbLf
txt = txt & "Размер - " & FSize & vbLf
txt = txt & "Ширина ячейки - " & CWidth & vbLf
txt = txt & "Высота ячейки - " & CHeight & vbLf
txt = txt & "Соотношение сторон - " & CScale
End With
' Звуковое приветствие
Application.Speech.Speak ("Hello, glad to see you!")
' Отображение окна с сообщением
MsgBox txt, vbInformation, "Информация об активной ячейке"
' Завершающая озвучка
Application.Speech.Speak ("Good bye!")
End Sub
```

Пример использования конструкции If … Then … Else

```
Sub WhoAreYou()
' Имя пользователя
Dim name As String
' Название для окна
Dim title As String
title = "Служба знакомств"
' Будем знакомиться?
Result = MsgBox("Хотите познакомиться?", vbYesNo, title)
' Проверка результата
If Result = vbYes Then
  name = InputBox("Как Вас зовут?", title)
' Если не указано имя, завершаем макрос
  If name = "" Then End
' Если имя указано, отображаем приветствие
  MsgBox "Добрый день, " & name & "! Приятно познакомиться!", vbOKOnly, title
  Else
' Знакомство не состоялось
    MsgBox "Очень жаль! Пока!", vbOKOnly, title
End If
End Sub
```

Пример использования оператора выбора Case и цикла For Each … Next

```
Sub SetColor()
' Переменная для обозначения ячейки
Dim Cell As Range
' Перебор ячеек в выделенном диапазоне
For Each Cell In Selection
With Cell
' Оператор выбора
  Select Case .Value
    Case 0
      .Interior.Color = vbYellow
      .Font.Bold = True
      .Font.Color = vbBlue
    Case Is > 0
      .Interior.Color = vbGreen
      .Font.Italic = True
      .Font.Color = vbWhite
    Case Is < 0
      .Interior.Color = vbRed
      .Font.Bold = True
      .Font.Italic = True
      .Font.Color = vbBlack
  End Select
End With
Next Cell
End Sub
```

Глава 11. Создание форм и обработка событий

Чтобы создать форму в VBE пройдите *Insert* –> *UserForm*. Добавьте в форму два текстовых окна и одну кнопку.

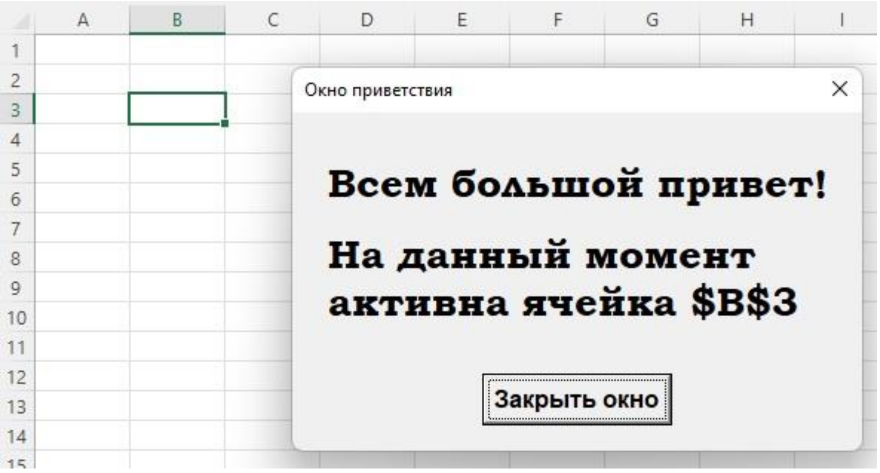

Рис. 10. Пользовательская форма

С формой связан код в отдельном Module2:

```
Sub SimpleFormShow()
With UserForm1.Label2
.Caption = .Caption & " " & ActiveCell.Address
End With
UserForm1.Show
End Sub
```

Свой код связан с кнопкой «Закрыть окно». Чтобы увидеть его дважды щелкните на кнопке:

```
Private Sub CommandButton1_Click()
Unload UserForm1
End Sub
```

Ключевое слово *Private* означает, что процедура доступна только для внутреннего вызова – вызвать эту процедуру как макрос из Excel не получится.

Обработка событий

Рассмотрим на примере формы для заполнения ячеек [числами Фибоначчи](). Первые два числа равны единице, а каждое следующее равно сумме двух предыдущих: 1, 1, 2, 3, 5, 8, 13, 21, ...

Программный код используемого нами макроса представлен в листинге 11.4.

Код макроса для заполнения ячеек числами Фибоначчи разместим в Module3

```
Sub Fibonacci()
' Отображение формы
UserForm2.Show
End Sub
```

В макросе всего одна команда, которая отображает окно формы. Основной код связан с элементами формы.

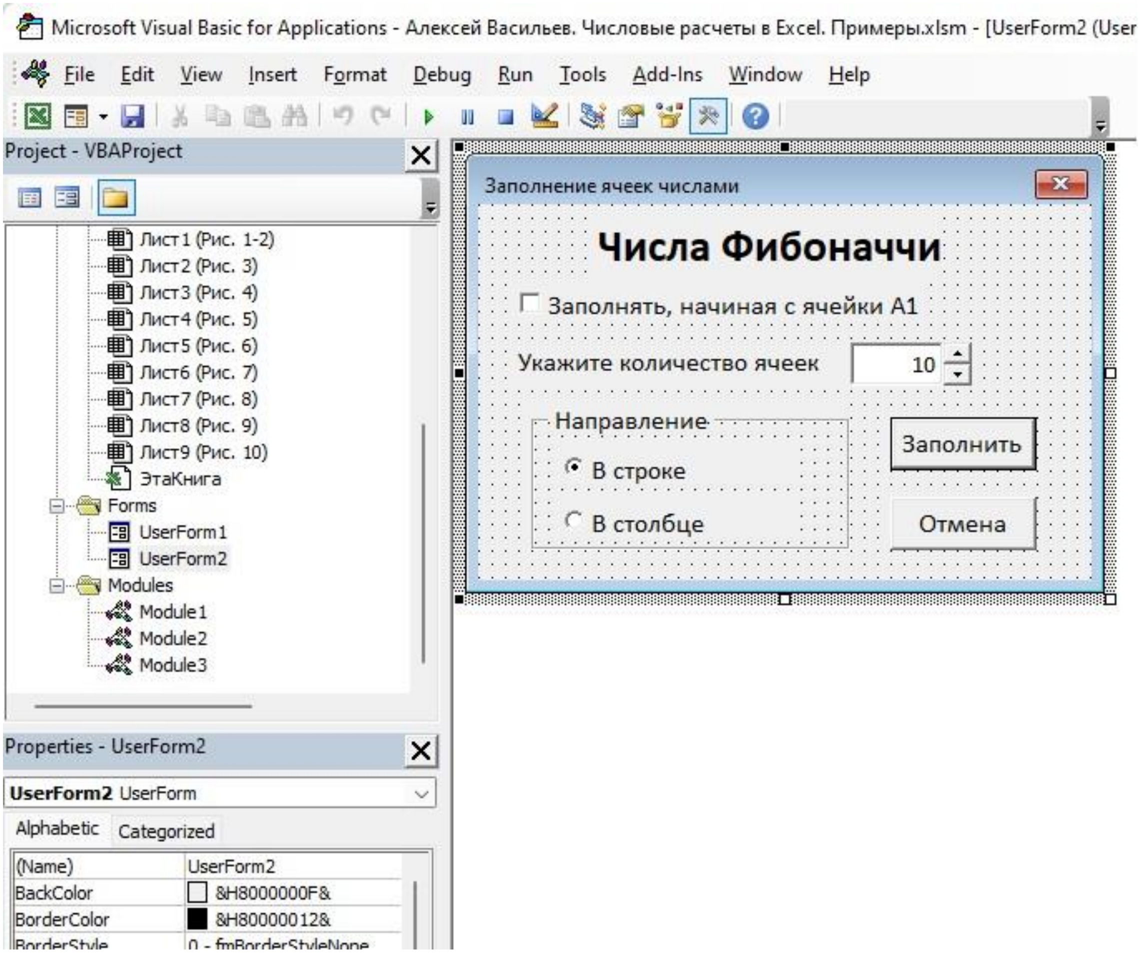

Рис. 11. UserForm2

Рассмотрим элементы формы. Свойство Caption UserForm2 = *Заполнение ячеек числами*. Текстовая метка Label1: Caption = *Числа Фибоначчи*. Далее следует элемент CheckBox1. Свойство Caption = *Заполнять, начиная с ячейки A1*. Свойство Value оставляем False (оно такое по умолчанию). Свойство Caption текстовой метки Label2 = *Укажите количество ячеек*. Справа от нее расположены элементы TextBox1 (Value = 10, Text Align = 3-fmTextAlignRight) и SpinButton1 (Value = 10, Min = 3, Max = 20). Далее Frame1. Свойство Caption = *Направление*. Внутри два переключателя: OptionButton1 (Caption = *В строке*, Value = True) и OptionButton2 (Caption = *В столбце*, Value = False). Справа две кнопки: CommandButton1 (Caption = *Заполнить*, Default = True) и CommandButton2 (Caption = *Отмена*, Default = False).

Теперь перейдем к программированию кнопок. Начнем с самого простого кода для кнопки закрытия окна формы. Для этого дважды кликните на ней. И введите код:

```
Private Sub CommandButton2_Click()
' Выгрузка формы из памяти
Unload UserForm2
End Sub
```

Далее щелкните на числовом переключателе – SpinButton1:

```
Private Sub SpinButton1_Change()
' Синхронизация текстового поля со значением числового переключателя
TextBox1.Text = SpinButton1.Value
End Sub
```

При щелчке на пиктограмме числового счетчика, в зависимости от того, какая из двух стрелок «щелкнута», значение переключателя (значение свойства Value) увеличивается/уменьшается на величину, определяемую свойством SmallChange (в нашем случае это единица) — если новое значение не выходит за допустимые пределы (задаются свойствами Max и Min). Процедура-обработчик запускается каждый раз при щелчке на счетчике. Поэтому, если мы изменяем состояние счетчика, это автоматически скажется на состоянии текстового поля.

Обработчик для текстового поля по умолчанию реагирует на событие/процедуру TextBoxl_Change. Нам это не подходит. Мы не хотим считать событием любое изменение содержимого поля (в процессе редактирования содержимого). Нас интересует ситуация, когда пользователь закончил редактировать поле. На этот случай предусмотрена процедура с ключевым словом AfterUpdate:

```
Private Sub TextBox1_AfterUpdate()
' Переменная для запоминания значения в текстовом поле
Dim num As Integer
' Считывание числового значения текстового поля
num = Val(TextBox1.Text)
With SpinButton1
' Проверка попадания значения в допустимый диапазон
If num >= .Min And num <= .Max Then
' Корректное значение присваивается числовому переключателю
.Value = num
Else
' Если значение некорректное, в текстовое поле
' вписывается значение числового переключателя
TextBox1.Text = .Value
End If
End With
End Sub
```

На момент начала редактирования текстового поля свойство Text текстового поля TextBox1 и свойство Value числового счетчика SpinButton1 одинаковы. Следующий шаг— пользователь ввел в текстовое поле новое значение – теперь это значение свойства Text объекта TextBox1. Мы его проверяем, и, если оно попадает в допустимый интервал значений, присваиваем соответствующее значение свойству Value числового счетчика SpinButton1. Если же новое значение в текстовом поле не попадает в допустимый интервал значений, в текстовое поле необходимо вернуть старое значение. Про это значение «помнит» свойство Value числового счетчика SpinButton1. Здесь действие обратное – свойству Text текстового поля TextBox1 присваивается значение свойства Value числового счетчика SpinButton1.

В теле процедуры TextBoxl_Afterllpdate() объявляется целочисленная переменная num, которой затем командой num = Val(TextBox1.Text) присваивается значение. Мы использовали функцию Val(), которой в качестве аргумента передается текстовое представление числа, а функция в качестве результата возвращает реальное число. Текстовое представление числа считывается инструкцией TextBox1.Text из текстового поля. Таким образом, в переменную num записано новое значение в текстовом поле. Затем на сцену выходит условный оператор, в котором проверяется условие принадлежности значения переменной num интервалу от SpinButton1.Min до SpinButton1.Max. При этом мы использовали логический оператор And. Если условие выполнено, свойству SpinButton1.Value присваивается значение num. В противном случае свойству TextBox1.Text присваивается значение SpinButton1.Value.

Код кнопки CommandButton1:

```
Private Sub CommandButton1_Click()
' Размер массива для чисел Фибоначчи
Dim N As Integer, i As Integer
' Объявление динамического массива
Dim F() As Integer
' Размер массива задается значением в текстовом поле
N = Val(TextBox1.Text)
' Определение размера массива
ReDim F(1 To N)
' Первые два числа в последовательности Фибоначчи
F(1) = 1
F(2) = 1
' Заполнение массива числами Фибоначчи
For i = 3 To N
F(i) = F(i - 1) + F(i - 2)
Next i
' Переменная для ссылки для начальной ячейки
Dim StartCell As Range
' Определение начальной ячейки
If CheckBox1.Value Then
Set StartCell = Range("A1")
Else
Set StartCell = ActiveCell
End If
' Определение "направления"заполнения
If OptionButton1.Value Then
For i = 1 To N
' Заполнение строки ячеек
StartCell.Offset(0, i - 1).Value = F(i)
Next i
Else
' Заполнение столбца ячеек
For i = 1 To N
StartCell.Offset(i - 1, 0).Value = F(i)
Next i
End If
' Выгрузка формы из памяти
Unload UserForm2
End Sub
```

В коде важный идеологический момент связан с тем, что мы сначала создаем массив из чисел Фибоначчи, а затем копируем числа из массива в ячейки рабочего листа. Поэтому предварительно определяются размеры массива (количество заполняемых ячеек), и этот массив заполняется числами. Затем мы определяем начальную ячейку заполнения и «направление» заполнения — в строке или в столбце.

Целочисленная переменная N объявляется для записи в нее размера массива. Целочисленная переменная I объявляется для использования в качестве счетчика цикла. Командой Dim F() As Integer объявляется динамический массив F() целых чисел. Затем из текстового поля инструкцией N = Val(TextBox1.Text) считывается количество заполняемых ячеек (размер массива). Чтобы выделить место под массив соответствующего размера, выполняем инструкцию ReDim F(1 To N).

Существует два типа массивов – статические и динамические. Размер статического массива должен быть известен на момент компиляции кода. Размер динамического массива определяется в процессе выполнения программы. Мы имеем дело со вторым случаем. С практической точки зрения разница между статическим и динамическим массивами в том, что размеры статического массива – константы. Например, командой Dim MyArray(1 To 100) As Integer объявляется целочисленный массив MyArray из 100 элементов (индексы элементов массива изменяются от 1 до 100). Мы же сначала инструкцией Dim F() As Integer декларируем, что F – это массив из целых чисел, неизвестно пока какого размера. А командой ReDim F(1 To N) подтверждаем, что массив из N элементов. Предварительно, разумеется, мы присваиваем значение переменной N.

Командами F(1) = 1 и F(2) = 1 заполняются первые два элемента массива. Для заполнения прочих элементов массива используем оператор цикла с командой F(i) = F(i-1)+F(i-2) (i – индексная переменная) в теле оператора цикла (каждый следующий элемент равен сумме двух предыдущих).

Следующий этап связан с определением начальной ячейки. Для этого командой Dim StartCell As Range мы объявляем переменную StartCell типа Range. Это будет ссылка на начальную ячейку. В условном операторе проверяем, установлен ли флажок опции CheckBox1 (значение свойства Value равно True или False). В зависимости от результата проверки выполняется команда Set StartCell = Range("A1") (начальная ячейка A1) или Set StartCell = ActiveCell (начальная ячейка – активная на момент запуска макроса). При присваивании значений объектным ссылкам используется ключевое слово Set.

Направление заполнения определяем путем проверки значения свойства OptionButtonl1.Value (выясняем, установлен переключатель OptionButton1 или нет). Если переключатель установлен, то запускается оператор цикла с командой StartCell.Offset(0,i-1).Value = F(i) в теле цикла. В противном случае запускается другой оператор цикла, в котором выполняется команда StartCell.Offset(i-1,0).Value = F(i). Этими командами ячейки рабочего листа заполняются значениями из массива F.

Метод Offset() вызывается из объекта ячейки и имеет два аргумента: смещение вдоль столбца и вдоль строки. В качестве результата возвращается ссылка на соответствующую ячейку. Так, например, инструкция StartCell.Offset(2,5).Value ссылается на ячейку, которая находится на 2 строки вниз и 5 столбцов вправо по отношению к ячейке StartCell. Функция Offset() в VBA подобна функции СМЕЩ в Excel.

Выгрузка формы из памяти осуществляется командой Unload UserForm2. Проверяем, как работает форма.

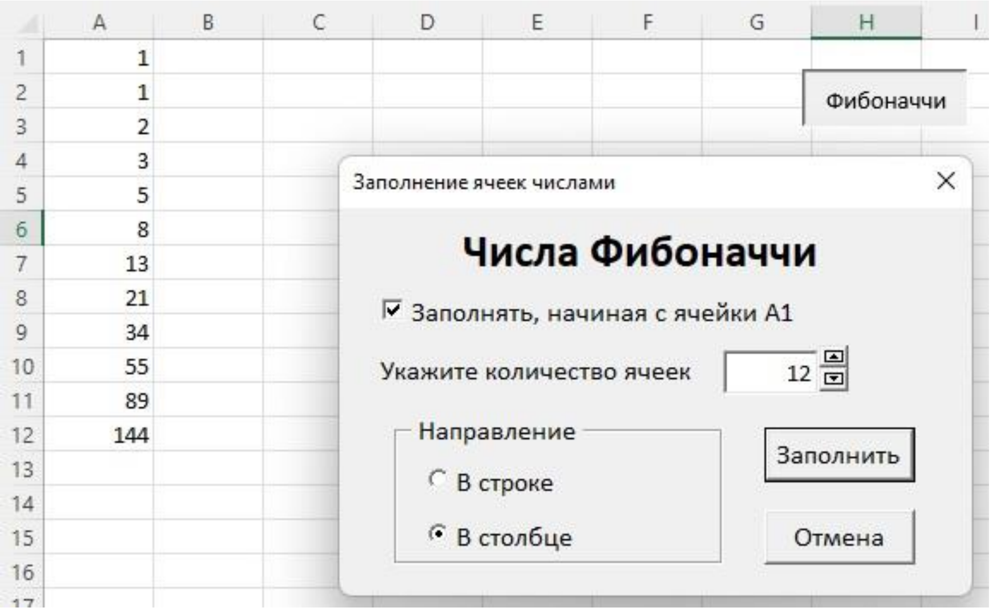

Рис. 12. Числа Фибоначчи

Глава 12. Решение алгебраических уравнений и систем

Показано, как использовать методы половинного деления (этот метод я изложил подробнее в отдельной заметке), хорд, касательных, последовательных приближений, Ньютона, градиентного спуска, а также методы линейной алгебры.

Глава 13. Интерполирование и аппроксимация

Представлены интерполяционный полином Лагранжа (см. подробнее), интерполяционный полином Ньютона, полиномиальная интерполяция, интерполяция набором функций, интерполяция сплайнами, а также методы аппроксимации.

Методы аппроксимации. Теория

Задача аппроксимирования близка по постановке к задаче интерполирования. В обоих случаях по набору «экспериментальных» данных необходимо восстановить «наилучшее» функциональное выражение между зависимой и независимой переменными. Отличие между этими задачами состоит в том, что в задаче интерполирования накладывается условие совпадения в узловых точках табличных значений и значений интерполяционной функции. При аппроксимировании параметров для варьирования недостаточно для того, чтобы аппроксимирующая функциональная кривая в каждой узловой точке принимала предопределенное значение. Ищется функция, которая будет иметь «наименьшее отклонение» от значений в узловых точках. При этом необходимо задать «метрику», через которую определяется «цена» отклонения от значения в узловой точке. В качестве метода оптимизации часто используют метод наименьших квадратов.

Предположим, имеется выражение $F(x, a_1, a_2, \ldots, a_m)$, на основе которого мы хотим аппроксимировать зависимость, заданную значениями $y_1, y_2, \ldots, y_m$ в узловых точках $x_1, x_2, \ldots, x_m$. Задача состоит в том, чтобы на основе критерия вычислить параметры оптимизации $a_1, a_2, \ldots, a_m$, которые входят в выражение для функции $F(x, a_1, a_2, \ldots, a_m)$.

В данном случае важно то, что количество m варьируемых параметров в аппроксимирующей функции не превышает количество n узловых точек, т.е. $m \leq n$. Причем, если имеет место равенство $m = n$, то речь идет о задаче интерполирования.

В качестве критерия мы будем использовать метод наименьших квадратов. Для этого построим функцию, определяемую как сумма квадратов разности «экспериментальных» и «теоретических» значений в узловых точках:

$$(9)\ \Phi(a_1, a_2, \ldots, a_m) = \sum_{k=1}^{n} (y_k - F(x_k, a_1, a_2, \ldots, a_m))^2$$

Суть метода состоит в том, чтобы подобрать такие параметры $a_1, a_2, \ldots, a_m$, при которых значение выражения $\Phi(a_1, a_2, \ldots, a_m)$ будет минимальным. В результате получаем систему из m алгебраических трансцендентных уравнений, решая которую, находим параметры $a_1, a_2, \ldots, a_m$.

Если учесть явное выражение для функции $\Phi(a_1, a_2, \ldots, a_m)$, уравнения системы для определения оптимизационных параметров могут быть записаны как

$$(10)\ \sum_{k=1}^{n} (y_k - F(x_k, a_1, a_2, \ldots, a_m)) \frac{\partial F(x_k, a_1, a_2, \ldots, a_m)}{\partial a_p} = 0$$

Индекс $p = 1, 2, \ldots, m$.

Часто аппроксимирующая функция представима в виде линейной комбинации базисных функций, и оптимизация выполняется по коэффициентам этой линейной комбинации — имеется в виду, что функция $F(x, a_1, a_2, \ldots, a_m) = a_1 f_1(x) + a_2 f_2(x) + \ldots + a_m f_m(x)$, где базисные функции $f_1(x), f_2(x), \ldots f_m(x)$ известны и определены однозначно.

Нередко задачу с нелинейной по параметрам оптимизации аппроксимирующей функцией удается свести к линейному случаю. Например, если по точкам (x_k, y_k) ($k = 1, 2, \ldots, n$) мы пытаемся вычислить оптимальные параметры для функции $F(x, A, b) = A\exp(bx)$, то можем рассмотреть набор точек $(x_k, \ln(y_k))$ и аппроксимировать эту зависимость линейной функцией $\ln(F(x, A, b)) = \ln A + bx$.

В этом случае мы можем значительно продвинуться в вопросе выбора оптимальных параметров $a_1, a_2, \ldots, a_m$. В частности, несложно понять, что

$$(11)\ \frac{\partial F(x_k, a_1, a_2, \ldots, a_m)}{\partial a_p} = f_p(x)$$

Тогда система для определения оптимизационных параметров записывается как

$$(12)\ \sum_{k=1}^{n}\left(y_k-\sum_{s=1}^{m}a_s f_s(x_k)\right)f_p(x_k)=0$$

для индексов p = 1, 2, ..., m. После преобразований эти уравнения можем записать в виде

$$(13)\ \sum_{s=1}^{m}\left(\sum_{k=1}^{n}f_p(x_k)f_s(x_k)\right)a_s=\sum_{k=1}^{n}f_p(x_k)y_k$$

Относительно параметров a_s (индекс s = 1, 2, ..., m) — это линейная неоднородная система алгебраических уравнений, которая достаточно легко решается, в том числе и в Excel. Ее удобно записать в матричном виде. Для этого введем ряд обозначений: вектор значений табулированной функции в узловых точках

$$(14)\ \vec{y}=[y_1,y,\dots,y_n]^T$$

… матрица $\hat{F}$ размерами mxn (m строк и n столбцов) значений базисных функций в узловых точках с элементами $F_{ij}=f_i(x_j)$ (индексы i = 1,2, ..., m и j = 1, 2, ..., n). Тогда систему уравнений для определения оптимизационных параметров можно в матричном виде записать как

$$(15)\ \hat{\mathbf{F}}\cdot\hat{\mathbf{F}}^T\vec{a}=\vec{\mathbf{F}}\cdot\vec{y}$$

где через

$$(16)\ \vec{a}=[a_1,a,\dots,a_n]^T$$

обозначен вектор, составленный из параметров оптимизации. Решение этой системы можем записать в виде

$$(17)\ \vec{a}=(\hat{\mathbf{F}}\cdot\hat{\mathbf{F}}^T)^{-1}\cdot\vec{\mathbf{F}}\cdot\vec{y}$$

где индекс -1 обозначает обратную матрицу.

Аппроксимация в Excel

Реализуем описанную выше схему на листе Excel. Оттолкнемся от исходных данных (область B2:K3) и построим аппроксимирующую зависимость на основе полиномиального выражения второй степени. Аппроксимирующая функция имеет вид $F(x)=a_1+a_2x+a_3x^2$, и задача состоит в определении по исходным данным параметров a_1, a_2 и a_3).

	A	B	C	D	E	F	G	H	I	J	K
1	№	1	2	3	4	5	6	7	8	9	10
2	X_n	0,0	0,1	0,3	0,6	0,7	0,9	1,3	1,5	1,9	2,0
3	Y_n	1,1	1,4	1,5	1,9	2,3	2,2	1,7	0,8	-0,8	-0,9
4	$f_1(x)$	1	1	1	1	1	1	1	1	1	1
5	$f_2(x)$	0,0	0,1	0,3	0,6	0,7	0,9	1,3	1,5	1,9	2,0
6	$f_3(x)$	0,0	0,0	0,1	0,4	0,5	0,8	1,7	2,3	3,6	4,0

	M	N	O	P	Q	R	S	T
3		1	2	3		a		b
4	1	10,0	9,3	13,3		1,03		11,20
5	2	9,3	13,3	21,7		3,01		5,41
6	3	13,3	21,7	38,0		-2,03		1,93

	A	B
8	x	f(x)
9	0,0	1,0
10	0,1	1,3
11	0,2	1,5
12	0,3	1,7
13	0,4	1,9
14	0,5	2,0
15	0,6	2,1
16	0,7	2,1
17	0,8	2,1
18	0,9	2,1
19	1,0	2,0
20	1,1	1,9
21	1,2	1,7

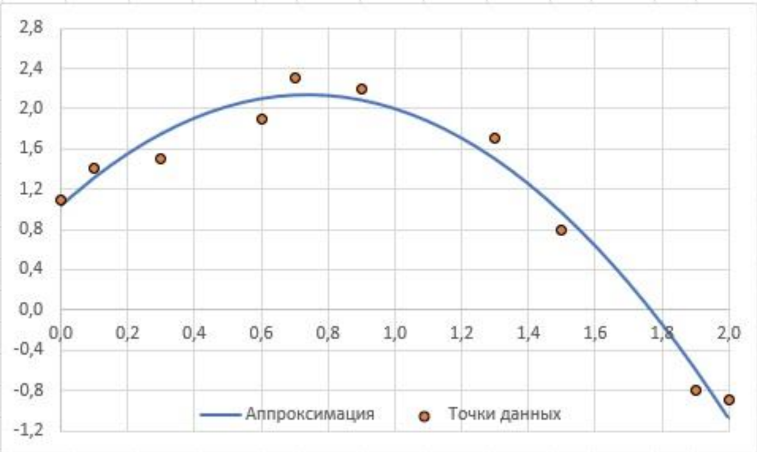

Рис. 13. Построение аппроксимирующей зависимости

Построим матрицу, составленную из значений базисных функций в узловых точках. Учитывая вид аппроксимирующей функции $F(x) = a_1 + a_2x + a_3x^2$, понятно, что базисными функциями являются $f_1(x) = 1$, $f_2(x) = x$ и $f_3(x) = x^2$. Значения этих функций в узловых точках вычислено в ячейках B4:K6. Это и есть та матрица, на основе которой вычисляются параметры аппроксимирующей функции. Выполним промежуточные расчеты:

- в ячейках N4:P6 введем формулу массива =МУМНОЖ(B4:K6;ТРАНСП(B4:K6)). Результатом вычислений является матрица $\hat{\mathbf{F}} \cdot \hat{\mathbf{F}}^T$ коэффициентов линейной системы уравнений;
- в ячейках T4:T6 введём формулу массива =МУМНОЖ(B4:K6;ТРАНСП(B3:K3)). Результатом является произведение $\vec{\mathbf{F}} \cdot \vec{y}$, представляющее собой вектор правой части линейной системы уравнений.

Для вычисления параметров аппроксимирующей функции в ячейки R4:R6 вводим формулу массива =МУМНОЖ(МОБР(N4:P6);T4:T6). Этой формулой реализуется выражение (17). Получаем вектор-столбик с коэффициентами a. Теперь значение аппроксимирующей функции в любой точке x можно вычислить по формуле $F(x) = 1{,}03 + 3{,}01x – 2{,}03x^2$ (диапазон A8:B29).

Глава 14. Дифференцирование и интегрирование

Рассмотрено: вычисление производной в узловых точках. Общие подходы к числовому интегрированию. Метод Чебышева. Метод Гаусса. Вычисление несобственных интегралов. Вычисление повторных интегралов.

Глава 15. Решение дифференциальных и интегральных уравнений

Метод последовательных приближений. Метод степенных рядов. Метод Рунге–Кутты. Метод Адамса. Метод Милна. Системы дифференциальных уравнений. Интегральные уравнения.

Глава 16. Вместо заключения

Рассмотрен бесплатный аналог Excel – приложением Open Office Calc.